Game Tree Searching By MIN/MAX Approximation : Ronald Rivest, 1988

Concepts Introduced

Generalized Mean (P-Mean)

This paper introduces the concept of Generalised Mean (P-Mean) of a list of positive real numbers and a non-zero real number P. It has the property that for large positive and negative values of P, Mp(ai) is a good approximation to max i (ai) or min i (ai), respectively. Unlike Max or Min which are discrete values Mp(ai) is a continuous value and it has derivatives w.r.t each ai. This can be used for a sensitivity analysis, i.e it can identify in an interesting way that leaf in a game tree upon whose value the value at the root depends most strongly. This is done by taking derivatives of the generalized mean value functions at each node and using the chain rule.

Game Tree Search

Given a finite tree C of configurations with root s. For small values of C, the tree can be explored completely and so optimal play is possible. On slightly larger trees MiniMax and AlphaBeta Pruning can produce optimal play even though a small fraction of the tree is explored, but for most interesting games the Game-Tree is so large that heuristic approximations are required. The heuristic method is usually a static evaluation function that gives an estimate v^(c) of the backed-up value v(c) for a nonterminal node c. The proposed technique in this paper uses a single evaluation function whereas other techniques such as the B* algorithm uses two evaluation functions for the lower and upper bound

Different Game Tree Search Techniques

Iterative Deepening returns the move determined by the last successful completed search at a depth ‘d’ before time runs out. This paper uses a different class of heuristic, ‘the Iterative heuristic’ which grow the search tree one step at a time. At each step a tip node (or leaf) of the current tree is chosen, and the successors of that tip node are added to the tree. Then the values provided by the static evaluator at the new leaves are used to provide new backed-up values to the leaves' ancestors. Such iterative techniques have been used in B* Algorithm and a couple of others and is not an original proposal of this paper.

Penalty based Iterative Search
The paper presents a general method for choosing which leaf to expand in an iterative method; And proposes a technique which is a specific instance of this method. The general idea is to assign a nonnegative "penalty" (or "weight") to every edge in the game tree such that edges representing bad moves arc penalized more than edges representing good moves

Proposed Technique : Min/Max Approximation

The "min/max approximation" heuristic is special case of the penalty-based search method, where the penalties are defined in terms of the derivatives of the approximating functions.
It proposes to use a relatively large P value and computes the approximation v~(c) of v^(c) for each c in E (a partial game tree). With idea that for large P the values v~(c) are a good approximation of v^(c).

While the decision on which node to pick next for expansion is a difficult one to answer in the conventional framework, this paper proposes to take the derivates with respect to each argument

Let D(x,y) = Delta(v~(x))/Delta(v~(y))

where y is any node in Ex, the subtree of E rooted at x. Thus D(s, c) measures the sensitivity of the root value v~(s) to changes in the tip value v~(c). Where 's' denotes the root node and 'c' is any child. The idea is to expand next the Tip c which has the maximum value of D(s,c), thus reducing the uncertainty in v~(s) in the most efficient manner. It is shown that the Tip x with largest value D(s,x) is the one with least penalty.

The idea of choosing the Tip c with maximum D(s,c) is formulated as a penalty based heuristic.

Summary of Results

While one can make an argument that its better to compute the accurate generalised mean (ignoring its computational cost) as that might give most accurate result for backed-up estimate v^(s) at the root, but they show that its better to use the min/max approximation v~(s).

When comparing with MiniMax and AlphaBeta Pruning, they show that for a for the Connect-Four game, they show that for time based resource limits alpha-beta seems to be superior to their implementation of the min/max approximation approach. However, if we base the comparison on move-based resource limits, the story is reversed: min/max approximation is definitely superior.

We note that the number of distinct positions considered by alpha-beta was approximately three times larger than the number of distinct positions considered by min/max when a time bound was in effect. When a move bound was in effect the number of distinct positions considered by each strategy was roughly equal; the fragmentation lossage (inefficiency due to throwing away the last partial search before timeout) of alpha-beta seemed to equal the inefficiencies of the min/max routine having to redescend the tree for each expansion.

Several other improvements were proposed to the implementation of the technique including parallelization which were not done as part of the paper.